

# EFEKTIVITAS WEB API DALAM INTEGRASI BAHASA PEMROGRAMAN MULTI PLATFORM

Astri Hijratul Rakhmah<sup>1</sup>, Hendro Purwoko<sup>2</sup>

<sup>1</sup>Politeknik LP3I Jakarta  
Gedung sentra Kramat Jl. Kramat Raya No. 7-9 Jakarta Pusat  
[astri.hijratul.lp3i@gmail.com](mailto:astri.hijratul.lp3i@gmail.com)

<sup>2</sup>Universitas Indraprasta PGRI  
Jl. Raya Tengah No.80, Gedong, Pasar Rebo, Jakarta Timur  
[hendroprwk08@gmail.com](mailto:hendroprwk08@gmail.com)

## ABSTRAK

Pertumbuhan teknologi komputer semakin cepat khususnya pada bahasa pemrograman. Saat ini dalam pembuatan Sistem Informasi tak cukup hanya menggunakan satu bahasa pemrograman saja, hal ini terjadi guna memenuhi kebutuhan teknologi yang makin bervariasi dengan tujuan yang berbeda pula. Banyak kasus pada organisasi yang awalnya menggunakan sistem informasi dengan bahasa pemrograman tertentu harus melakukan penyesuaian dan migrasi dengan biaya yang tak murah. Sebagai solusi menghindari permasalahan tersebut diperlukan aplikasi yang menjembatani berbagai macam *platform* dan bahasa pemrograman dalam sistem informasi yang telah berjalan, sehingga dibutuhkan API. API atau Application Programming Interface merupakan konsep yang bekerja pada lapisan *logic* dan *database* sehingga dapat digunakan untuk menjembatani sistem informasi ketika berkomunikasi antar perangkat lunak dan *platform* yang berbeda. Akses API menggunakan protokol HTTP, sehingga disebut dengan Web API agar mudah diakses kapan dan di mana pun selama terhubung dengan internet. Penelitian ini mengusung penggunaan Web API yang dibangun dengan bahasa pemrograman PHP dan pada sisi *client* akan menggunakan dua teknologi yaitu .NET dan HTML yang berkolaborasi dengan JQuery menggunakan alur hidup bermodel Waterfall diharapkan dapat membangun Web API yang kuat dan bebas kesalahan.

Kata Kunci: Multi Platform, Web API, Waterfall

## ABSTRACT

*The growth of computer technology is increasingly fast in programming languages. Currently, the information system is not enough to use only one programming language, this happens in the need of technology that is increasingly varied with different purposes. Many cases in organizations that originally used information systems with certain programming languages should make adjustments and migrations at an inexpensive cost. As a solution to avoid such problems is necessary to bridge various platforms and programming languages in the system of information that has been running, so it needs an API. An API or Application Programming Interface is a concept that works on a layer of logic and a database so that it can be used to bridge information systems when communicating between different software and platforms. API Access uses the HTTP protocol, so it's called the Web API to be easily accessible anytime and anywhere while connected to the Internet. This research brings the use of Web APIS built with PHP programming language and on the client side will use two technologies namely .NET and HTML that collaborate with JQuery using a waterfall in the model, is expected to build a strong Web API and error-free.*

*Keyword: Multi Platform, Web API, Waterfall*

## PENDAHULUAN

Teknologi merupakan cara manusia dalam menangani keterbatasannya. Seperti halnya sejarah perkembangan komputer yang awalnya berasal dari alat hitung sederhana kemudian menjadi alat hitung mekanis hingga berkembang dengan mengalami banyak sekali tahapan. Teknologi umumnya dibangun melalui

kode atau bahasa pemrograman, beberapa puluh tahun lalu mungkin yang kita tahu bahasa pemrograman Pascal, C, Delphi, Visual Basic dan lainnya. Akan tetapi, karena kebutuhan informasi semakin meningkat, maka bahasa pemrograman pun turut mengalami perkembangan dan bermunculan bahasa pemrograman baru.

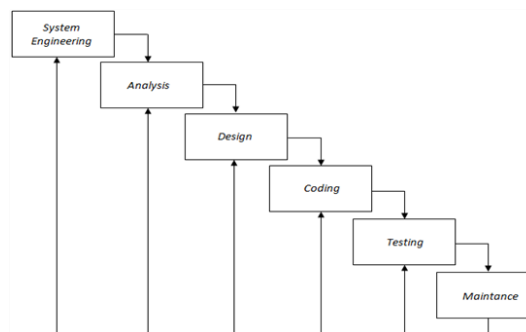
Banyak kasus dialami perusahaan yang awalnya menggunakan sistem informasi dalam bahasa pemrograman tertentu harus melakukan penyesuaian, bahkan melakukan migrasi hingga membutuhkan banyak waktu dan sumber daya, sehingga hal ini memerlukan aplikasi yang dapat menjembatani atau menyatukan berbagai macam platform dan bahasa pemrograman pada lingkup sistem informasi tersebut.

API merupakan cara yang tepat untuk memecahkan permasalahan diatas. API merupakan singkatan dari *Application Programming Interface* memungkinkan para developer membuat koneksi atau tautan antara dua jenis perangkat lunak yang berbeda. API diakses menggunakan protokol HTTP ( Web ) agar dapat digunakan oleh berbagai macam jenis client seperti aplikasi mobile, aplikasi Web, dan aplikasi Desktop yang dapat membantu perusahaan (Erick Kurniawan, 2014) sehingga disebut sebagai Web API. Untuk pertukaran data disampaikan menggunakan HTTP dengan serialisasi JSON (Warsito et al., 2017).

Pada tulisan ini pembuatan Web API menggunakan bahasa pemrograman PHP pada sisi server, sedangkan disisi *Client* menggunakan dua teknologi, yaitu berbasis Desktop dengan menggunakan bahasa pemrograman C# dan Web dengan menggunakan bahasa pemrograman HTML yang didukung dengan JQuery, keduanya akan bekerja secara Asynchronous sehingga proses komunikasi dengan Web API terasa ringan dan tak mengalami *bottle necking*. Selain menggunakan teknologi yang telah diuraikan diatas, penelitian ini menggunakan metode Waterfall sebagai alur hidup pengembangannya dengan tujuan meminimalisir kesalahan dari Web API yang dibangun.

#### METODE PENELITIAN

Adapun pembangunan sistem informasi pada penelitian ini menggunakan daur hidup sistem bermodel Waterfall agar setiap tahap mendapatkan hasil yang maksimal.



Gambar 1. Model Waterfall

Tahapan tersebut disusun menjadi tujuh tahap dengan uraian sebagai berikut:

1. Analisa Kebutuhan Sistem: Dalam tahap ini penulis akan melakukan persiapan kebutuhan yang akan digunakan selama penelitian. Kebutuhan yang dimaksud adalah persiapan *software* atau *platform* sesuai yang telah ditetapkan.
2. Analisa Kebutuhan Integrasi: Mengetahui Web API apa yang paling sesuai untuk digunakan dalam integrasi sistem. Selain itu juga tahap ini digunakan untuk mencari tahu bagaimana dan apa saja tahapan yang dibutuhkan untuk melakukan integrasi (*scripting* dan *setting*).
3. Perancangan Basis Data: Melakukan desain database, mendefinisikan kebutuhan tabel dalam *database* dan *field* apa saja yang akan digunakan dalam aplikasi yang dibuat.
4. Perancangan Interface (Antar Muka): Perancangan dibuat menggunakan Balsamiq Mockup
5. *Coding Platform 1*: membuat sistem CRUD atau *Create Read Update Delete* menggunakan Bahasa pemrograman PHP dengan *database* MySQL yang diletakkan pada server
6. *Coding Platform 2*: sebagai *client* dengan dua basis yang berbeda, yaitu Desktop berbahasa program C# dan Web dengan JQuery
7. Integrasi: Tahap akhir merupakan tahap penggabungan sisi *client* dan server yang dibuat di masing-masing platform bahasa pemrograman dengan Web API.

## HASIL DAN PEMBAHASAN

Agar lebih mudah dibahas, maka diambil contoh proses menambah, ubah dan hapus data kegiatan yang didalamnya terdapat database bernama “dbevent”, tabel yang diberi nama “event” dan server yang digunakan berbasis Linux.

Tabel 1. Struktur Tabel event

Kolom	Tipe Data & Ukuran	Keterangan
id event	Integer	Primary Key, Otomatis
Event	Varchar (255)	
Description	Long Text	
Date	Date	
Time	Time	
input_time	Datetime stamp	Tanggal dan jam otomatis

Pada kode PHP dibuat Class DB yang berfungsi sebagai operator *database* untuk operasi: simpan, ubah, hapus dan menampilkan data serta setiap *request* dan *response* yang terdapat pada objek JSON disesuaikan dengan materi apa yang akan ditampilkan pada aplikasi dan web (Rulloh et al., 2017). Adapun class tersebut dituliskan sebagai berikut:

```
<?php
class DB{
    var $cn;
    public function connect(){
        $this->conn = mysqli_connect ("localhost", "xyz", "abc",
        "dbevent")
        or die(mysqli_error());
    }

    public function run($s){
        mysqli_query($this->cn, $s) or die (mysqli_error($this-
        >conn));
    }

    public function read($s){
        $arr = Array();
        $q = mysqli_query($this->cn, $s) or die (mysqli_error($this-
        >cn));

        while($baris = mysqli_fetch_assoc($q)){
            $arr[] = $baris;
        }
        return $arr;
    }
}
?>
```

API dibuat melalui kode PHP yang berfungsi menerima, mengolah dan melakukan operasi database. PHP mengambil data yang dikirim dari client dengan `$_REQUEST`. Pengiriman data dikirim melalui URL yang berada pada protokol HTTPS dengan menggunakan pola seperti contoh:

<https://www.namadomain.com/event.php?act=1&ev=PKK&dsk=Kegiatan%20warga&tgl=03-26-2019&jam=09:00>

Perhatikan kode yang berhuruf tebal, URL yang dikirim *client* pada event.php disisipkan data berupa: variable act, bernilai 1 yang ditentukan dengan act = 1, ev = PKK, dsk = Kegiatan rutin, tgl = 03-26-2019 dan jam = 09:00. URL tersebut ditangkap dengan `$_REQUEST` dan terbentuk kode seperti dibawah ini:

```
<?php
include "DB.php";

$act = null;

if (!isset($_REQUEST['act'])){
    $act = null;
}else{
    $act = $_REQUEST['act'];
}

$db = new DB();
$db->connect();

$res = array();

if ($act == null){
    $res = array("res"=>"0");
    print json_encode($res);
}elseif($act == "1"){
    $sql = "insert into event (event, description, date, time) "
        . "values (" . $_REQUEST['ev'].","
        . $_REQUEST['dsk']."," . $_REQUEST['tgl'].","
        . $_REQUEST['jam'].")";
    $db->run($sql);

    $arr[] = array(
        "res" => $_REQUEST['event']. " inserted"
    );

    $res = array("res"=>$arr);
    print json_encode($res);
}elseif($act == "2"){
    $sql = "update event set event = " . $_REQUEST['event'].","
        . "description = " . $_REQUEST['deskripsi'].","
        . "date = " . $_REQUEST['tgl'].","
        . "time = " . $_REQUEST['jam'].","
        . "where id = " . $_REQUEST['id'];
    $db->run($sql);

    $arr[] = array(
        "res" => $_REQUEST['ev']. " updated"
    );

    $res = array("res"=>$arr);
    print json_encode($res);
}elseif($act == "3"){
    $sql = "delete from event where id = " . $_REQUEST['id'];
    $db->run($sql);

    $arr[] = array(
        "res" => $_REQUEST['id']. " deleted"
    );

    $res = array("res"=>$arr);
    print json_encode($res);
```

```
}elseif($act == "4"){
    $sql = "select * from event";

    $res = array("res"=>$d->read($sql));
    print json_encode($res);
}elseif($act == "5"){
    $sql = "select * from event where event like '%" .
        $_REQUEST['find'] . "%'";

    $res = array("res"=>$d->read($sql));
    print json_encode($res);
}
?>
```

Keluaran dari kode diatas akan menampilkan data berformat JSON, dari kode diatas pada bagian **\$act == 4** berfungsi menampilkan data *event* seperti dibawah ini:

```
{
  "res": [
    {
      "id": "1",
      "event": "Workshop Dasar Pemrograman Web
      ASP.NET",
      "description": "",
      "date": "2019-05-25",
      "time": "13:00:00",
      "input": "2019-05-12 03:21:53"
    },
    {
      "id": "8",
      "event": "Android ",
      "description": "",
      "date": "2019-05-25",
      "time": "09:00:00",
      "input": "2019-05-12 03:23:28"
    }
  ]
}
```

Pada sisi *client* bahasa pemrograman C# akan mengirimkan data dengan kode berikut:

```
using (var httpClient = new HttpClient()) {
    try{
        var json = await httpClient.GetStringAsync(
            "https://www.namadomain.com/event.php?action=2" +
            "&event=" + HttpUtility.UrlEncode(tb_event.Text)+
            "&deskripsi="
            +HttpUtility.UrlEncode(tb_deskripsi.Text) +
            "&tgl=" + tgl +
            "&jam=" + jam);
    } catch (HttpRequestException ex) {
        tb_log.Text = ex.ToString();
    }
}
```

Setiap data yang dikeluarkan melalui JSON akan ditampilkan kedalam Grid, ada beberapa tahapan untuk menampilkannya: pertama meminta data melalui API, kemudian JSON yang diterima diolah menjadi *data table* dan menerapkannya pada Grid dengan kode ini.

```
using (var httpClient = new HttpClient()){
    var json = await
    httpClient.GetStringAsync("https://www.namadomain.com/eve
    nt.php?action=5&find=" + tb_cari.Text);

    JObject data = JObject.Parse(json);
    String values = data["result"].ToString();
    dt_event = (DataTable)JsonConvert.DeserializeObject(values,
        (typeof(DataTable)));
}
if (dt_event.Rows.Count == 0) return;
dataGridView1.DataSource = dt_event;
```

*Client* selanjutnya dengan pemrograman berbasis web dengan dukungan JQuery yang bekerja di belakang HTML. Proses komunikasi dengan Server sama dengan C#, yaitu dengan API. Untuk menyimpan data maka kode yang digunakan sebagai berikut:

```
$("#submit").click(function(){
    var event = $("#event").val();
    var desk = $("#desk").val();
    var tgl = $("#tgl").val();
    var jam = $("#jam").val();

    var myUrl =
    "https://www.namadomain.com/event.php?action=1&event="
    + event + "&deskripsi=" + desk + "&tgl=" + tgl + "&jam=" + jam;

    $.ajax({
        url: myUrl,
        success: function(result){
            loadGrid();
        }
    });
});
```

Sedangkan untuk menampilkannya membutuhkan bantuan "Data Table" yang dibuat dengan JQuery agar lebih menarik dan data yang ditampilkan terasa ringan. Adapun kode yang ditulis sudah terbentuk *function* agar mudah dipanggil kapan pun, kode tersebut adalah:

```
var myUrl =
    "https://www.namadomain.com/event.php?action=4";

$.ajax({
    url: myUrl,
    dataType: "json",
    success: function(result){
        var rows, json = result.result;
        var table =
        $("#table").DataTable( {
            "language": {
                "emptyTable": "Tak ada Data"
            },
            data: json,
            columns: [
                { data: 'event' },
                { data: 'deskripsi' },
                { data: 'date' },
                { data: 'time' },
            ],
        });
```

```
}  
});
```

### SIMPULAN DAN SARAN

Penggunaan API dalam sistem informasi dinilai efektif dan mudah dikembangkan hingga masa mendatang tanpa perlu risau dengan kondisi platform dan bahasa pemrograman yang berbeda. Pada bahasan terlihat bahwa satu API dapat menjadi “pelayan” bagi *platform* lain. Daur hidup sistem Waterfall yang diusung pun juga mempermudah langkah kerja yang terarah.

### DAFTAR PUSTAKA

Erick Kurniawan. (2014). Implementasi Rest Web Service Untuk Sales Order Dan Sales Tracking Berbasis Mobile. *Jurnal EKSIS*, 7(1), 1–12.

<https://media.neliti.com/media/publications/78853-ID-implementasi-rest-web-service-untuk-sale.pdf>

Rulloh, A., Mahmudah, D. E., & Kabetta, H. (2017). Implementasi REST API pada Aplikasi Panduan Kepaskibraan Berbasis Android. *Teknikom: Teknologi Informasi, Ilmu Komputer Dan Manajemen*, 1(2), 85–89. <http://journal.swu.ac.id/index.php/teknikom/article/view/50>

Warsito, A. B., Ananda, A., & Triyanjaya, D. (2017). Penerapan Data JSON Untuk Mendukung Pengembangan Aplikasi Pada Perguruan Tinggi Dengan Teknik Restfull Dan Web Service. *Technomedia Journal*, 2(1), 26–36. <https://doi.org/10.33050/tmj.v2i1.313>